

MISSION : DÉPLOYER DES PIPELINES D'INTÉGRATION CONTINUE

L'intégration continue (CI) consiste à automatiser l'intégration des modifications apportées au code par plusieurs développeurs dans une base de code unique. Il s'agit d'une pratique de développement logiciel où les développeurs livrent fréquemment leur travail dans le dépôt central de code (Github ou Stash). Il existe ensuite des outils automatisés qui construisent le code nouvellement livré et effectuent une révision du code, etc. selon les besoins lors de l'intégration. Le CI/CD (Intégration continue et Déploiement continu) permet aux organisations de livrer des logiciels rapidement et efficacement. Le CI/CD facilite un processus efficace pour mettre les produits sur le marché plus rapidement que jamais, livrer continuellement du code en production et assurer un flux continu de nouvelles fonctionnalités et de corrections de bogues via la méthode de livraison la plus efficace.

Objectif:

L'objectif de cette mission est d'expérimenter la configuration de pipelines d'intégration continue

Consignes :

- Il est recommandé de former une équipe SLAM/SISR pour la réalisation de cette mission
- Cette mission fait l'objet d'un compte rendu au format PDF (Mission-Jenkins-NOM-Prénom.pdf) qui fera apparaître, selon le cas, une réponse à la question ou une capture d'écran de la commande et son résultat.

Partie 1 : Mise en place d'un serveur sur Ubuntu

Objectif : Vous êtes invité à créer un nouveau serveur Ubuntu 22.04 et à suivre quelques étapes pour augmenter la sécurité et la praticabilité de votre serveur.

1. Installez une machine virtuelle Ubuntu Server 22.04

```
Site : https://ubuntu.com/download/server  
Ubuntu 64 bits  
RAM : 8192 MB  
HD : 32 GB  
Processeur 2 CPU
```

2. Connectez-vous en tant qu'utilisateur root
3. Créer un nouvel utilisateur
4. Par défaut sur Ubuntu 22.04, les utilisateurs qui sont membres du groupe sudo sont autorisés à utiliser la commande sudo. En tant que root, ajouter votre nouvel utilisateur au groupe sudo
5. Les serveurs Ubuntu 22.04 peuvent utiliser le pare-feu UFW pour s'assurer que seules les connexions à certains services sont autorisées. Examinez la liste des profils UFW installés en tapant :

```
ufw app list
```
6. Autoriser la connexion SSH sur le pare-feu UFW
7. Activer le pare-feu UFW
8. Vérifier le statut et les règles de l'UFW

Partie 2 : Mise en place de Java sur le serveur linux

La machine virtuelle Java, appelée JVM, est une machine de calcul abstraite ou une interface de machine virtuelle qui pilote le code Java.

Java et la JVM (la machine virtuelle de Java) sont nécessaires pour de nombreux types de logiciels

Objectif : Vous êtes invité à installer Java sur votre nouveau serveur et éventuellement sélectionner la version que vous souhaitez utiliser pour vos projets.

1. Comprendre la différence entre JRE et JDK et celle entre OpenJDK et Oracle JDK

```
OpenJDK contrairement à Oracle JDK, est l'implémentation open-source de
Java et est fourni avec Ubuntu.
Le JDK fournit les outils logiciels essentiels pour développer en Java,
tels qu'un compilateur et un débogueur. Le JRE est utilisé pour exécuter
les programmes Java.
```

2. Mettez à jour votre index des paquets apt
3. Vérifier si Java est déjà installé

```
java -version
```

4. Vous pouvez avoir besoin du JDK en plus du JRE afin de compiler et d'exécuter certains logiciels spécifiques basés sur Java. Pour installer le JDK, exécutez la commande suivante, qui installera également le JRE.

```
sudo apt install default-jdk
```

5. Vérifiez que le JDK est installé en contrôlant la version de javac, le compilateur Java
6. Vous pouvez avoir plusieurs installations Java sur le même serveur. Configurez la version par défaut à utiliser sur la ligne de commande à l'aide de la commande update-alternatives.

Selection	Path	Priority
0	/usr/lib/jvm/java-11-openjdk-amd64/bin/java	1111
auto mode		
1	/usr/lib/jvm/java-11-openjdk-amd64/bin/java	1111
manual mode		

```
* 2          /usr/lib/jvm/java-11-oracle/bin/java          1091
manual mode

Press <enter> to keep the current choice[*], or type selection number:
```

7. Avec la commande précédente, notez et copiez où java est localisé sur votre serveur ubuntu.

```
Dans l'exemple précédent, OpenJDK 11 se trouve à l'adresse
/usr/lib/jvm/java-11-openjdk-amd64/bin/java
```

8. Ouvrez /etc/environment en utilisant nano ou votre éditeur de texte préféré

```
sudo nano /etc/environment
```

À la fin de ce fichier, ajoutez la ligne suivante, en vous assurant de remplacer le chemin surligné par votre propre chemin copié, et de ne pas inclure la partie bin/ du chemin :

```
JAVA_HOME="/usr/lib/jvm/java-11-openjdk-amd64"
```

En modifiant ce fichier, vous définirez le chemin JAVA_HOME pour tous les utilisateurs de votre système. Enregistrez le fichier et quittez l'éditeur.

9. Rechargez maintenant ce fichier pour appliquer les changements à votre session actuelle :

```
source /etc/environment
```

10. Vérifiez que la variable d'environnement est définie en affichant le JAVA_HOME

```
echo $JAVA_HOME
```

Partie 3 : Mise en place de Jenkins sur le serveur linux

Jenkins est un outil open source d'automatisation des processus d'intégration continue et de déploiement continu. Sa principale utilisation réside dans la création de pipelines CI/CD. Il permet aux utilisateurs de se connecter sur des systèmes de gestion de codes sources tels que GitHub ou des outils de conteneurisation comme Docker et Kubernetes. D'autres opérations sont possibles et plusieurs plug-ins sont disponibles afin qu'il puisse prendre en charge de nombreux autres outils.

Objectif : Vous êtes invité à installer et configurer Jenkins sur votre serveur

1. La version de Jenkins incluse dans les paquets Ubuntu par défaut est souvent en retard sur la dernière version disponible du projet lui-même. Pour vous assurer de disposer des derniers correctifs et fonctionnalités, utilisez les paquets maintenus par le projet pour installer Jenkins. Pour ce faire, ajoutez la clé du référentiel à votre système :

```
wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key |sudo  
gpg --dearmor -o /usr/share/keyrings/jenkins.gpg
```

2. Dans la commande précédente, gpg --dearmor est utilisée pour convertir la clé dans un format reconnu par apt. Ajoutez l'adresse du dépôt de paquets Debian au fichier sources.list du serveur :

```
sudo sh -c 'echo deb [signed-by=/usr/share/keyrings/jenkins.gpg]  
http://pkg.jenkins.io/debian-stable binary/ >  
/etc/apt/sources.list.d/jenkins.list'
```

The [signed-by=/usr/share/keyrings/jenkins.gpg] portion of the line ensures that apt will verify files in the repository using the GPG key that you just downloaded.

3. Mettez à jour le système pour que apt utilise le nouveau référentiel
4. A présent, Installez Jenkins et ses dépendances

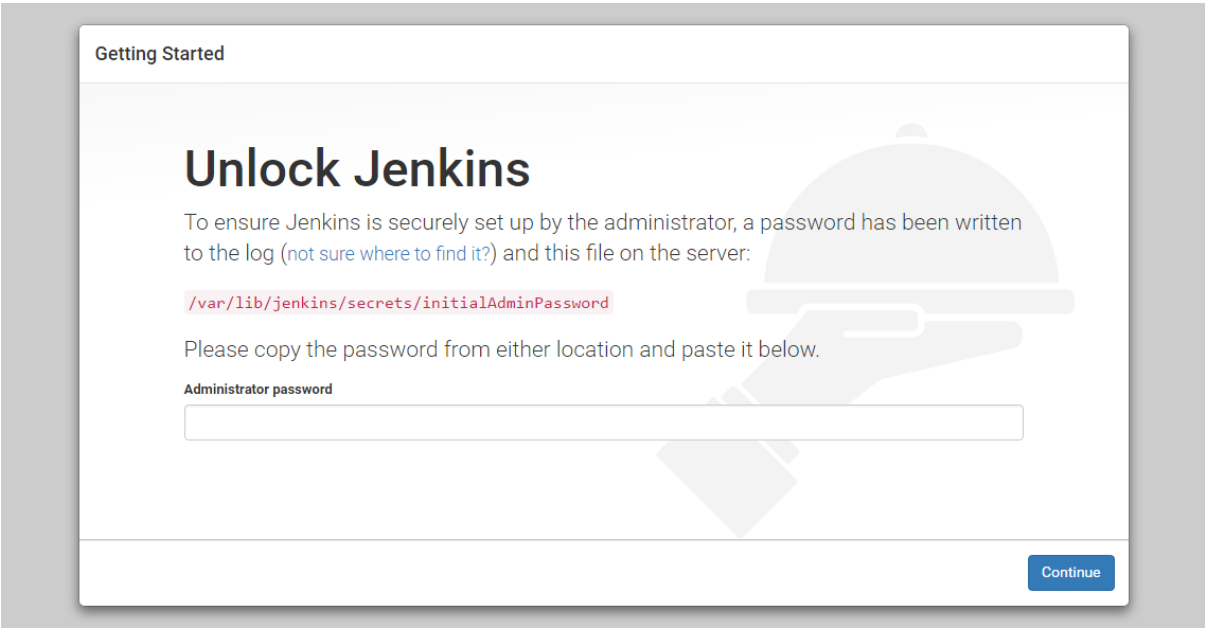
```
sudo apt install jenkins
```

5. Démarrer jenkins en utilisant la commande systemctl
6. Utilisez la commande status de systemctl pour vérifier que Jenkins a bien démarré
7. Par défaut, Jenkins fonctionne sur le port 8080. Ouvrez ce port sur le pare-feu UFW
8. Vérifier le statut et les règles de l'UFW pour confirmer que le port est bien ouvert

9. Pour configurer votre installation, visitez Jenkins sur son port par défaut, 8080, en utilisant le nom de domaine ou l'adresse IP de votre serveur :

```
http://ip_de_mon_server:8080
```

Vous devriez recevoir l'écran Déverrouiller Jenkins, qui affiche l'emplacement du mot de passe initial.

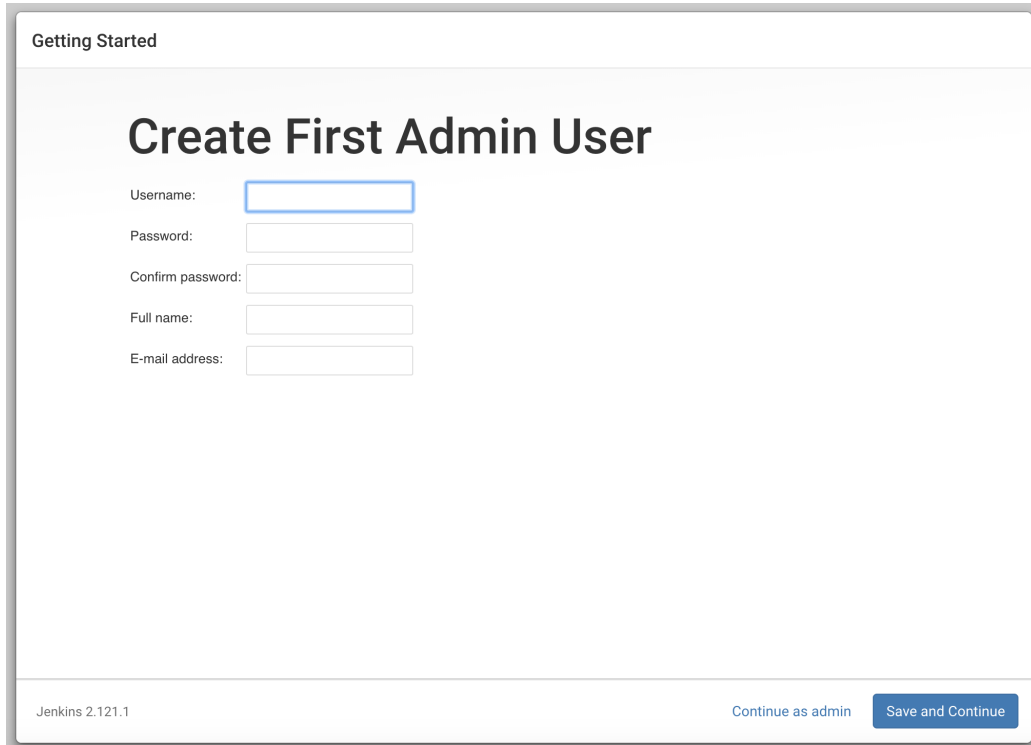


10. Dans la fenêtre du terminal, utilisez la commande cat pour afficher le mot de passe. Copiez le mot de passe alphanumérique à 32 caractères du terminal et collez-le dans le champ du mot de passe administrateur, puis cliquez sur Continuer.
11. L'écran suivant présente l'option d'installer les plugins suggérés ou de sélectionner des plugins spécifiques. Nous allons cliquer sur l'option Installer les plugins suggérés, ce qui va immédiatement lancer le processus d'installation.

Getting Started

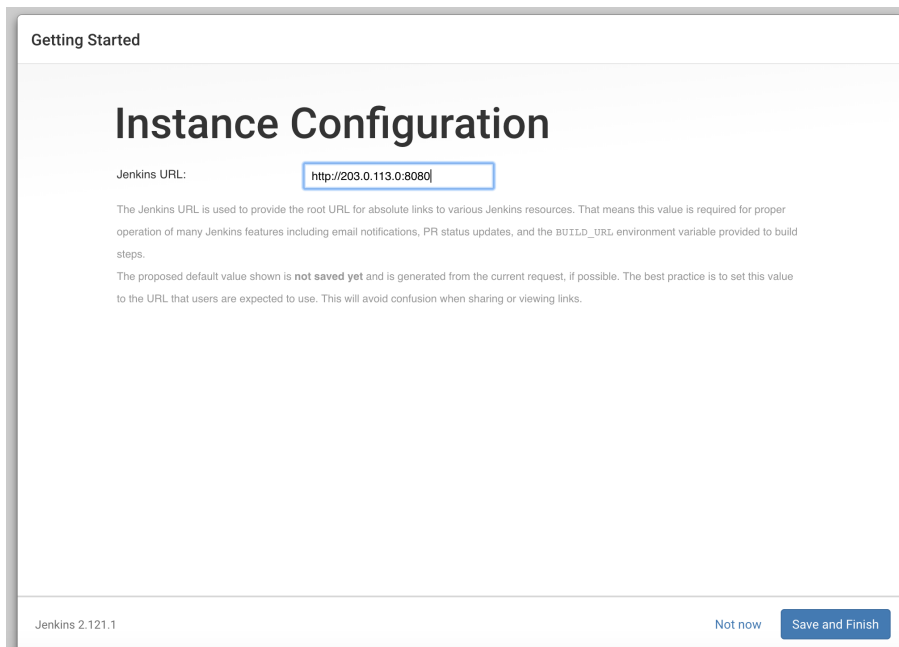
✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding	<pre>** Pipeline: Milestone Step ** JavaScript GUI Lib: jQuery bundles (jQuery and jQuery UI) ** Jackson 2 API ** JavaScript GUI Lib: ACE Editor bundle ** Pipeline: SCM Step ** Pipeline: Groovy ** Pipeline: Input Step ** Pipeline: Stage Step ** Pipeline: Job ** Pipeline Graph Analysis ** Pipeline: REST API ** JavaScript GUI Lib: Handlebars bundle ** JavaScript GUI Lib: Moment.js bundle Pipeline: Stage View ** Pipeline: Build Step ** Pipeline: Model API ** Pipeline: Declarative Extension Points API ** Apache HttpComponents Client 4.x API ** JSch dependency</pre>
✓ Timestamper	✓ Workspace Cleanup	✓ Ant	✓ Gradle	
🔗 Pipeline	🔗 GitHub Branch Source	🔗 Pipeline: GitHub Groovy Libraries	✓ Pipeline: Stage View	
🔗 Git	🔗 Subversion	🔗 SSH Slaves	🔗 Matrix Authorization Strategy	
🔗 PAM Authentication	🔗 LDAP	🔗 Email Extension	🔗 Mailer	

12. Une fois l'installation terminée, il vous sera demandé de configurer le premier utilisateur administratif. Il est possible de sauter cette étape et de continuer en tant qu'administrateur en utilisant le mot de passe initial ci-dessus, mais prenez un moment pour créer l'utilisateur.




The screenshot shows the 'Getting Started' page of Jenkins 2.121.1. The main heading is 'Create First Admin User'. Below it, there are five input fields: 'Username:', 'Password:', 'Confirm password:', 'Full name:', and 'E-mail address:'. The 'Username' field is highlighted with a blue border. At the bottom right, there are two buttons: 'Continue as admin' and 'Save and Continue'. The version 'Jenkins 2.121.1' is displayed at the bottom left.


13. Vous obtiendrez une page de configuration d'instance qui vous demandera de confirmer l'URL préférée pour votre instance Jenkins. Entrez l'adresse IP de votre serveur.





The screenshot shows the 'Getting Started' page of Jenkins 2.121.1. The main heading is 'Instance Configuration'. Below it, there is a 'Jenkins URL:' label and an input field containing 'http://203.0.113.0:8080/'. Below the input field, there is a paragraph of text explaining the purpose of the Jenkins URL. At the bottom right, there are two buttons: 'Not now' and 'Save and Finish'. The version 'Jenkins 2.121.1' is displayed at the bottom left.

14. Cliquez sur Commencer à utiliser Jenkins pour visiter le tableau de bord principal de Jenkins. À ce stade, l'installation de Jenkins est terminée.


 **Jenkins**


 **1**


 **sammy**


 **log out**


Dashboard


 New Item


 People

 Build History

 Manage Jenkins

 My Views

 Lockable Resources

 New View

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job

→

Set up a distributed build

Set up an agent

→

Configure a cloud

→

Learn more about distributed builds

↔

Add description

REST API

Jenkins 2.332.2

8

Partie 4 : Mise en place de Nginx sur le serveur linux

NGINX est un logiciel open source pour le service web, le reverse proxying, la mise en cache, l'équilibrage de charge, le streaming multimédia, etc. Au départ, il s'agissait d'un serveur web conçu pour offrir des performances et une stabilité maximales.

Objectif : Vous êtes invité à installer Nginx sur votre serveur

1. Mettez à jour notre index local des paquets apt afin d'avoir accès aux listes de paquets les plus récentes.
2. Nginx est disponible dans les dépôts par défaut d'Ubuntu. Installez le paquet nginx en utilisant apt install. Si vous êtes invité à redémarrer des services, appuyez sur ENTER pour accepter les valeurs par défaut et continuer. apt installera Nginx et toutes les dépendances requises sur votre serveur.
3. Nginx s'enregistre en tant que service auprès du pare-feu UFW lors de l'installation, ce qui permet d'autoriser facilement l'accès à Nginx. Listez les configurations d'applications identifiées sur notre machine par UFW.

```
sudo ufw app list
```

```
Nginx Full : Ce profil ouvre à la fois le port 80 (trafic web normal, non crypté) et le port 443 (trafic crypté TLS/SSL).
Nginx HTTP : Ce profil ouvre uniquement le port 80 (trafic Web normal, non chiffré).
Nginx HTTPS : Ce profil n'ouvre que le port 443 (trafic crypté TLS/SSL).
```

4. Il est recommandé d'activer le profil le plus restrictif qui autorise toujours le trafic que vous avez configuré. Pour l'instant, nous n'avons besoin d'autoriser que le trafic sur le port 80.

```
sudo ufw allow 'Nginx HTTP'
```

5. Vérifier le statut et les règles de l'UFW pour confirmer que le port est bien ouvert
6. À la fin du processus d'installation de Nginx, Ubuntu le démarre. Vérifier avec la commande systemctl que le service Nginx est en cours d'exécution.

```
• nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2022-03-01 16:08:19 UTC; 3 days ago
```

7. Trouvez l'adresse IP de votre serveur.
8. Saisissez l'adresse IP de votre serveur dans la barre d'adresse de votre navigateur pour voir la page d'accueil Nginx par défaut.

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Partie 5 : Mise en place de Docker sur le serveur linux

Docker est un outil qui peut emballer une application et ses dépendances dans un conteneur isolé, qui pourra être exécuté sur n'importe quel serveur. Il ne s'agit pas de virtualisation, mais de conteneurisation, une forme plus légère qui s'appuie sur certaines parties de la machine hôte pour son fonctionnement. Les conteneurs vous permettent d'exécuter vos applications dans des processus isolés des ressources. Ils sont similaires aux machines virtuelles, mais les conteneurs sont plus portables, plus économes en ressources et plus dépendants du système d'exploitation hôte.

Objectif : Vous êtes invité à installer et configurer Docker Community Edition sur votre serveur

1. Bien comprendre la procédure

```
Le paquet d'installation de Docker disponible dans le dépôt officiel d'Ubuntu peut ne pas être la dernière version. Pour nous assurer d'obtenir la dernière version, nous allons installer Docker à partir du dépôt officiel de Docker. Pour ce faire, nous allons ajouter une nouvelle source de paquet, ajouter la clé GPG de Docker pour nous assurer que les téléchargements sont valides, puis installer le paquet.
```

2. Mettez à jour votre liste de paquets apt existante
3. Ensuite, installez quelques paquets prérequis qui permettent à apt d'utiliser les paquets via HTTPS :

```
sudo apt install apt-transport-https ca-certificates curl  
software-properties-common
```

4. Ajoutez ensuite la clé GPG du dépôt officiel de Docker à votre système :

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key  
add -
```

5. Ajouter le dépôt Docker aux sources APT.

```
sudo add-apt-repository "deb [arch=amd64]  
https://download.docker.com/linux/ubuntu focal stable"
```

Cela met également à jour notre base de données de paquets avec les paquets Docker du repo nouvellement ajouté.

6. Assurez-vous que vous êtes sur le point d'installer à partir du dépôt Docker au lieu du dépôt Ubuntu par défaut.

```
apt-cache policy docker-ce
```

- Est-ce que Docker-ce est installé ?
 - Quel est le numéro de version de Docker, candidat à l'installation ?
7. Installez le paquet docker-ce avec apt
 8. Docker devrait maintenant être installé, le démon démarré, et le processus activé pour démarrer au démarrage. Vérifiez qu'il est en cours d'exécution avec systemctl.
 9. Par défaut, la commande docker ne peut être exécutée que par l'utilisateur root ou par un utilisateur du groupe docker, qui est automatiquement créé au cours du processus d'installation de Docker. Pour éviter d'avoir à taper sudo chaque fois que vous exécutez la commande docker, ajoutez votre nom d'utilisateur au groupe docker

```
sudo usermod -aG docker ${USER}
```

10. Pour appliquer la nouvelle composition du groupe, déconnectez-vous du serveur et reconnectez-vous, ou tapez ce qui suit :

```
su - ${USER}
```

- Expliquez la commande ci-dessus
11. Trouver la commande pour ajouter au groupe docker, un utilisateur sous lequel vous n'êtes pas connecté.

```
X usermod -aG docker Y
```

- Indice : Trouver X et Y pour trouver la commande complète
12. Utilisez la commande docker info pour afficher des informations sur l'ensemble du système concernant l'installation de Docker.
 13. Les conteneurs Docker sont construits à partir d'images Docker. Par défaut, Docker extrait ces images de Docker Hub, un registre de Docker géré par Docker, la société à l'origine du projet Docker. Tout le monde peut héberger ses images Docker sur

Docker Hub, de sorte que la plupart des applications et des distributions Linux dont vous aurez besoin auront des images hébergées à cet endroit.

- Expliquez les termes conteneur et image sous docker

14. Vérifiez que vous pouvez accéder et télécharger des images à partir de Docker Hub

```
docker run hello-world
```

- Avez-vous remarqué ?

Docker était initialement incapable de trouver l'image hello-world localement, il a donc téléchargé l'image depuis Docker Hub, qui est le dépôt par défaut. Une fois l'image téléchargée, Docker a créé un conteneur à partir de l'image et l'application dans le conteneur s'est exécutée, affichant le message.

15. Recherchez l'image Ubuntu parmi les images disponibles sur Docker Hub

```
docker search ubuntu
```

- Combien de résultats avez-vous ?

16. Téléchargez l'image officielle ubuntu sur votre ordinateur en utilisant la sous-commande pull

17. Comprenez-bien le fonctionnement

Une fois qu'une image a été téléchargée, vous pouvez alors exécuter un conteneur utilisant l'image téléchargée avec la sous-commande run. Comme vous l'avez vu dans l'exemple hello-world, si une image n'a pas été téléchargée lorsque docker est exécuté avec la sous-commande run, le client Docker téléchargera d'abord l'image, puis exécutera un conteneur en l'utilisant.

18. Visualisez les images qui ont été téléchargées sur votre ordinateur

```
docker images
```

- A savoir : les images que vous utilisez pour exécuter des conteneurs peuvent être modifiées et utilisées pour générer de nouvelles images, qui peuvent ensuite être téléchargées (poussées est le terme technique) vers Docker Hub ou d'autres registres Docker.

19. Exécutez un conteneur en utilisant la dernière image d'Ubuntu avec la sous-commande run.

```
docker run -it ubuntu
```

- A savoir : La combinaison des commutateurs -i et -t vous donne un accès interactif au shell dans le conteneur.

20. Quel est l'identifiant du conteneur dans l'invite de commande ?

```
Exemple Sortie  
root@d9b100f2f114:/#
```

21. Mettez à jour la base de données des paquets apt à l'intérieur du conteneur.

22. Installez à l'intérieur du conteneur, le paquet Node.js avec apt

23. Vérifiez que Node.js est installé à l'intérieur du conteneur

```
node -v
```

24. Toutes les modifications apportées à l'intérieur du conteneur ne s'appliquent qu'à ce conteneur. Tapez exit pour quitter le conteneur

25. Utilisez docker ps pour afficher les conteneurs actifs sur votre ordinateur.

26. Affichez tous les conteneurs (actifs et inactifs), en exécutant docker ps avec le commutateur -a.

27. Démarrez un conteneur arrêté en utilisant docker start, suivi de l'ID du conteneur ou du nom du conteneur.

28. Utilisez docker ps pour vérifier son statut

29. Arrêtez le conteneur en cours d'exécution en utilisant docker stop suivi de l'ID ou du nom du conteneur.

30. Après avoir installé Node.js dans le conteneur Ubuntu, vous avez maintenant un conteneur fonctionnant à partir d'une image, mais le conteneur est différent de l'image que vous avez utilisée pour le créer. Mais vous pourriez vouloir réutiliser ce conteneur Node.js comme base pour de nouvelles images plus tard. Faites un commit (livrez) des changements à une nouvelle instance d'image Docker en utilisant la commande suivante.

```
docker commit -m "Ce que vous avez fait à l'image" -a "Nom de l'auteur"  
container_id repertoire/nouveau_nom_image
```

Exemple :

```
docker commit -m "Ajout de Node.js" -a "euphraim" d9b100f2f636  
euphraim/ubuntu-nodejs
```

31. Lorsque vous faites un commit (livraison) d'une image, la nouvelle image est enregistrée localement sur votre ordinateur. Plus tard en classe, vous apprendrez comment pousser une image vers un registre Docker tel que Docker Hub afin que d'autres puissent y accéder.
32. Affichez à nouveau les images Docker pour voir la nouvelle image, ainsi que l'ancienne dont elle est dérivée

```
docker images
```

A savoir : Plus tard en classe, vous apprendrez également à créer des images à partir d'un fichier Dockerfile, ce qui vous permet d'automatiser l'installation de logiciels dans une nouvelle image.

Partie 6 : Expérimentez la mise en place CICD dans Jenkins